A learning and control approach based on the human neuromotor system

Brandon Rohrer
Cybernetic Systems Integration Department
Intelligent Systems and Robotics Center
Sandia National Laboratories
MS 1010, PO Box 5800
Albuquerque, NM 87185-1010, USA
brrohre@sandia.gov

Steven Hulet
Computer Science Department
Brigham Young University
Provo, UT, 84604, USA
stevehulet@gmail.com

***Abstract*— Current models of human motor learning and control typically employ continuous (or near continuous) movement commands and sensory information. However, research suggests that voluntary motor commands are issued in discrete-time submovements. There is also reasonable support for the hypothesis that human sensory experience is episodic as well. These facts have motivated the development of a learning algorithm that employs discrete-time sensory and motor control events, S-learning. We present this algorithm together with the results of simulated robot control. The results show that the learning that takes place is adaptive and is robust to a variety of conditions that many traditional controllers are not capable of handling, including random errors in the actuators and sensors, random transmission time delays, hard nonlinearities, time varying system behavior, and unknown structure of system dynamics. The performance of S-learning suggests that it may be an appropriate high-level control scheme for complex robotic systems, including walking, cooperative manipulation, and humanoid robots.**

***Index Terms*— adaptive control, machine learning, discrete-time, temporal difference learning**

I. Introduction

When mathematically modeling human motor learning and control, it is common to make a number of assumptions: 1) Sensory information and control information are usually considered to be continuous in time. 2) Perception and movements are often expressed in terms of fixed, external Cartesian coordinates. 3) In many cases, velocity, acceleration, and higher derivatives of position are explicitly represented in the motion planner. 4) Kinematic states are assumed to be sensed at high resolution. While models based on these assumptions can describe some aspects of human movement, none of these assumptions has been proven. In addition, these models are typically used only to model and predict a limited class of movement ([e.g. ballistic reaching movements [1]). In this paper, we propose an alternative motor learning model. This model employs as working assumptions that both motor commands and sensory information are passed in a discrete, episodic fashion, quantized in time.

Evidence for discrete-time motor commands, also known as submovements, is widespread and accounts for a large number of disparate phenomena in motor behavior. Observations of slow finger movements [2], eye saccades [3], tracing constant curvature paths [4], cyclical movements [5], [6], [7], infant reaching movements [8], ballistic movements [9], movements of recovering stroke patients [10], [11], and movements requiring high accuracy [12] are all consistent with a theory of submovements. The discrete-time nature of movement is evident not only in movement kinematics, but also in the electromyograph (EMG) signals of agonist and antagonist muscles [2].

Evidence for the discrete nature of sensory experiences is more subtle. The concept was originally proposed by William James [13] and more recently by Stroud [14]. One particularly striking phenomenon that suggests discrete sensory experience is the wagon wheel illusion under steady light. Due to the rapid series of photographs of which movies are composed, it is commonly observed that a spoked wagon wheel appears to rotate slowly backward while rolling rapidly forward. Interestingly, the same effect can also be observed in life (as opposed to motion pictures) under steady light [15], suggesting a periodic sampling mechanism in human vision. In another experiment, two lights that blinked with a slight delay were occasionally perceived to flash simultaneously [16], an occurrence that was suggested to be a function of the phase relationship with alpha (8–12 Hz) cortical rhythms [17]. Other observations that suggest discrete sensory experiences are the sharp dependence of perceived causality on delay times and periodicities in reaction times [18]. A more in-depth review of the case for discrete perception is made in Ref. [19].

II. S-learning

The problem of learning to interact with an unknown environment while having no explicit model of one's own dynamics is particularly challenging to address because of its generality. Yet human infants presented with the problem eventually manage to find a solution. Some previous work in this area specific to navigation is motivated and described in Ref. [20]. The algorithm we propose to address this problem more generally is a reinforcement learning algorithm [21], qualitatively similar to the temporal-difference learning algorithm known as Q-learning [22]. It also has a strong component of sequence learning, a tool

used to model, among other things, handwriting generation [23]. In contrast to common algorithms for sequence learning, such as Markov models and neural networks, the approach we present does not collapse previous experiences into a statistical compendium (see Ref. [24]). Rather, it maintains a library of repeatedly observed patterns that is referenced like a database. Due to the algorithm's emphasis on sequences, it will be referred to here as *S-learning*.

Like Q-learning, S-learning can create a model using only a raw series of inputs and does not require a reward function to be specified for each state. However, it differs from Q-learning and other implementations of reinforcement learning in that it does not require a fixed goal state. In Q-learning, the goal or reward states must remain constant for learning to take place. If the goal is changed, all previous experience is rendered obselete. S-learning, in contrast, addresses the time-varying-goal problem in reinforcement learning by "remembering" sequences of previous actions and sensory inputs. When a new goal is presented, the S-learning algorithm searches past sequences for series of actions that moved the system from its current state to the desired state. It can utilize past experience to reach a novel goal.

S-learning treats data categorically and has no explicit representation of distance between the inputs. Using inputs from the natural number line to illustrate, the number 2 would not be assumed to be closer to 1 than would the number $1,000,000$; each would be interpreted as categorically different than the others. (However, S-learning would quickly learn to associate 1 and 2 while making small steps along the number line.) While this hinders the performance S-learning in one sense by disregarding useful information, it also broadens the scope of problems S-learning can address to those incorporating categorical data, such as ice cream flavors, text input, emotional states, abstract concepts, etc. In terms of human behavioral modeling, S-learning provides a single tool that can model the learning and control of both human movement and human cognition.

In S-learning, learning occurs by repeated observation of sensory and control events. The block diagram in Fig. 1a describes the process in detail. Motor and sensory events of different magnitudes are binned and treated categorically; events falling into different categories are initially considered to be unrelated. That is, extrapolation and interpolation do not occur explicitly. During learning (as in an infant), the motor control system issues a set commands and observes the resulting sensory events. As patterns are observed repeatedly, they are recorded and extended. This growing library of patterns constitutes the motor controller's "experience base." A key feature of this algorithm is that it is able to "bootstrap" a partial model of its universe, based on whatever experiences it has recorded to date. It is not paralyzed by the fact that it hasn't already visited the entire state space.

In a more formal terms, consider the following definitions:

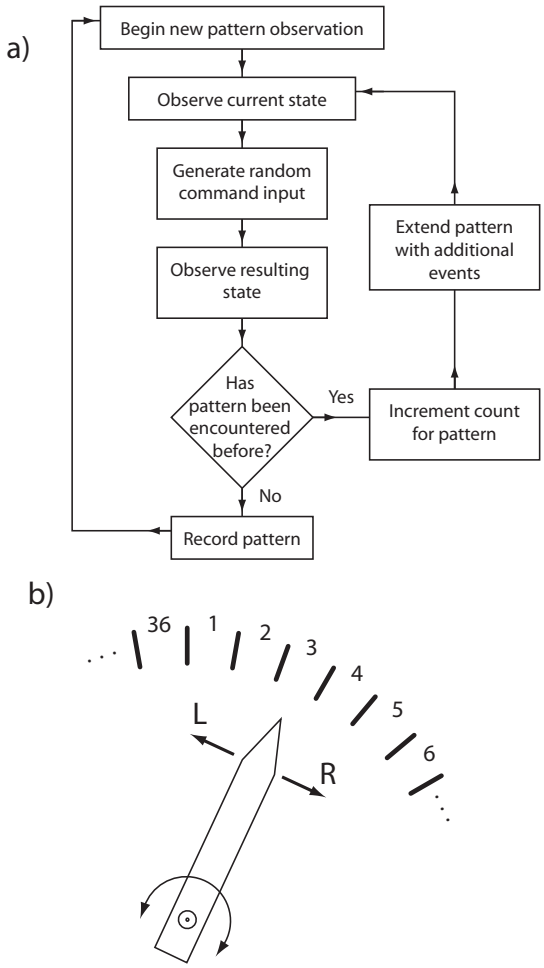

Fig. 1. a) S-learning algorithm. The S-learning agent identifies its dynamics and those of its environment by randomly generating patterns and noting repeated occurrences. It initially observes its current state, generates a random input, and then observes the resulting state. If it is a pattern that it has not encountered before, it records the pattern in memory and repeats the process. If the pattern has been previously observed, the agent notes the observation and then extends the pattern by generating another random input. In this way, patterns of increasing length are recorded as training progresses. **b)** The simulated rotary pointer robot showing 10° position sensing bins and R and L movement commands.

$$\theta = \{\chi_1, \chi_2, \chi_3, ...\} \tag{1}$$

Where χ is an *event* and θ is an ordered series of events. χ can be a sensory event or a command event, a raw data measurement or a symbol, depending on the inputs to the S-learning algorithm. θ may be finite or infinite and is hereafter referred to as the *event stream*. It represents the whole of the "experience" of the learning algorithm. *Patterns*, denoted by ρ, are finite series of events drawn from θ:

$$\rho_j = \{\chi_1^j, \chi_2^j, ..., \chi_n^j\} \tag{2}$$

A *pattern library*, λ, is an unordered set of previously observed patterns:

$$\lambda = \{\rho_1, \rho_2, \rho_3, ...\} \tag{3}$$

In addition, each library entry ρ_j has a counter associated with it, N_j, and a recency measure, ϕ_j.

The S-learning algorithm processes θ by matching its first several elements with a pattern from λ. The longest possible match, $\rho_M = \{\chi_1^M, ..., \chi_n^M\}$, is found for the first n elements of θ, the count, N_M, associated with the library entry for ρ_M is incremented by one, and the corresponding recency measure, ϕ_M is reset to zero. In addition, χ_{n+1} is appended to ρ_M to form a new pattern, ρ_{new}, that is then added to λ.

Initially $\lambda = \{\emptyset\}$. Patterns of increasing length are "grown" as they are repeatedly observed. The storing of actual patterns taken from θ, as opposed to a statistical representation of the patterns (as in a Markov model), allows order and context-specific information to be preserved. The gradual building of patterns is a means of limiting the memory requirements of the algorithm. Instead of storing a sparse N-dimensional matrix to track the occurrence of all patterns of length N, the non-zero entries are stored in list form. This allows the storage of long patterns with large sets of distinct events in reasonable memory. Periodic "forgetting" of very rarely observed patterns will further limit the storage requirements.

S-learning can be applied to any discretized and quantized stream of data. By using it to close a control loop, it can serve as a dynamic world modeler for an autonomous system. This scheme is depicted in Fig. 1a. The command-generation policy shown here is an exploratory one; it generates random commands and observes the results. More sophisticated exploration policies are possible. Modification of S-learning's command-generation policy to one of explicit goal-seeking transforms it from a randomly exploring learning into an algorithm for closed-loop control. The exploration policy can be replaced by a policy that searches through the agent's experience base to find the command that is most likely to lead to a desirable state. In this way, thoughtful selection of policy can yield highly sophisticated agent behavior.

III. Simulation

Consider an implementation of S-learning in a simulated rotary pointer robot (Fig. 1b). Possible sensory events for the pointer robot consist of position sensing in 10° bins, resulting in 36 distinct states (for convenience, numbered 1 through 36). Possible command events for the pointer are 10° rotation clockwise (R) and 10° rotation counterclockwise (L).

While the rotary pointer robot is a simple simulation that does not closely resemble the human neuromuscular system, it provides a generalized learning challenge. Beginning with no model of the robot's function, no implied structure, and no connection between neighboring sensor states poses a learning problem that can easily be transferred to systems with more degrees of actuation freedom, more sensors, and a greater range of actuator outputs. The rotary pointer robot problem posed in this way is representative of far more complex learning problems.

An example of how S-learning operates shows the simplicity of the approach. One sample excerpt of an event history resulting from random movements might consist of 3 R 4 R 5 L 4 R 5 L 4 L 3 L 2. The S-learning agent would break the event history into short patterns, 3 R 4, 4 R 5, 5 L 4, etc. When these patterns are encountered again in subsequent excerpts of the event history, they will be extended, producing patterns such as 3 R 4 R 5, 4 R 5 L 4, and 5 L 4 R 5 R 6.

A simulation of S-learning applied to the pointer robot system was implemented in C++. Six conditions were simulated:

Simple system. Measurement states 1–36 and command events R and L as described previously.

Hard stop. Same as the simple system, but with a "hard stop" inserted at 0°, between positions 36 and 1, prohibiting continuous rotational movement. This is an example of a hard nonlinearity in the environment, analogous to intermittent contact in manipulation.

Sensory state scramble. Same as the simple system, but after 5000 trials, the numerical labels for sensory states are reshuffled in random order. This is an extreme and discontinuous example of time variance. The system changes radically, making all prior learning inapplicable.

Command reversal. Same as the simple system, but after 5000 trials the commands "reverse"; an R command produces *counterclockwise* motion and an L command produces *clockwise* motion. This is another example of a dramatic change in system dynamics. In this case, an action has exactly the opposite of the intended effect.

Random error. Same as the simple system, but with up to 5° of random error added to each command event, resulting in movements of between 5° and 15°. With measurement resolution limited to 10°, the error will express itself as measurement states being either skipped or unchanged when a command is issued. The random variations are in essence a large, cumulative source of noise with a signal-to-noise ratio near 1.

Random delays. Same as the simple system, but each command event has a 50% chance of being delayed and executed at the instant the *next* command is issued. As a result, when a command is issued, zero, one, or two command events may actually take place. Non-determinacy in time is a feature of control across high-traffic networks, such as the World Wide Web.

In each case, the S-learning agent generated random command events and attempted to predict the results before executing the command. Predictions were generated by searching through previously observed patterns for instances containing a portion of the current event history. Patterns that provided the longest match with the event

history were selected. Among those, patterns that were observed recently or that had been observed many times were favored more highly. Once a pattern was selected, a prediction was obtained by reading "what happened next" when the situation had been encountered previously. In each condition, the S-learning agent began with a clean slate; that is, there were no previously observed experiences upon which to build. As a result, lack of prior experience made it impossible for the agent to offer a prediction in some cases. These were counted as unsuccessful predictions.

It is worth noting that the prediction tasks demonstrated here are nontrivial. The five conditions contained instances of hard nonlinearities, dramatic time variance, large stochastic movement error, and nondeterministic time delays, any one of which can impose insurmountable challenges for certain learning algorithms. However, they are challenges that the human motor learning mechanism routinely faces and successfully overcomes without difficulty. Taken together, they constitute something of a proving ground for any model of motor learning purporting to describe that of a human.

IV. Results

During simulation of each of the six conditions, the learning agent generated a database of patterns. Representative patterns observed were 16 R 17 R 18 R 17 L 18 R 19 (observed 6 times after 10,000 trials), 25 R 26 R 27 L 26 L 25 R 26 (observed 8 times), and 22 R 23 L 22 R 23 L 22 L 21 (observed 11 times). In the case of the simple system, a total of 2155 repeated patterns were observed, occupying 599 kilobytes of memory. The longest patterns observed included five movement events, a limit imposed by the software, rather than by the inherent function of the S-learning agent. On average, patterns contained between 3 and 4 movement events. Simulating 10,000 trials for one condition took approximately two minutes. Given that the simple system was learned within the first 2000 trials, only 24 seconds were required to learn the system's dynamics completely.

The results of the simulations are shown in Fig. 2. As shown in the plot, the S-learning agent achieved 100% accuracy in the simple system after 2000 trials. The S-learning agent showed similar performance in the presence of a hard stop. In both these conditions, the performance of the system is deterministic, allowing correct predictions at every time step.

Scrambling the sensory state labels changed the system fundamentally, making the probability of encountering a previously observed pattern small. Learning essentially began from scratch, and the initial learning transient was repeated after scrambling. Reversing the directions of the commands resulted in a marked decrease in performance initially, but the agent recovered within 4000 trials after that and predicted the last 1000 trials perfectly. It is likely that the "reversed" state took so much longer to learn than the "scrambled label" state because in the latter the S-learning agent's previous experience was completely inapplicable

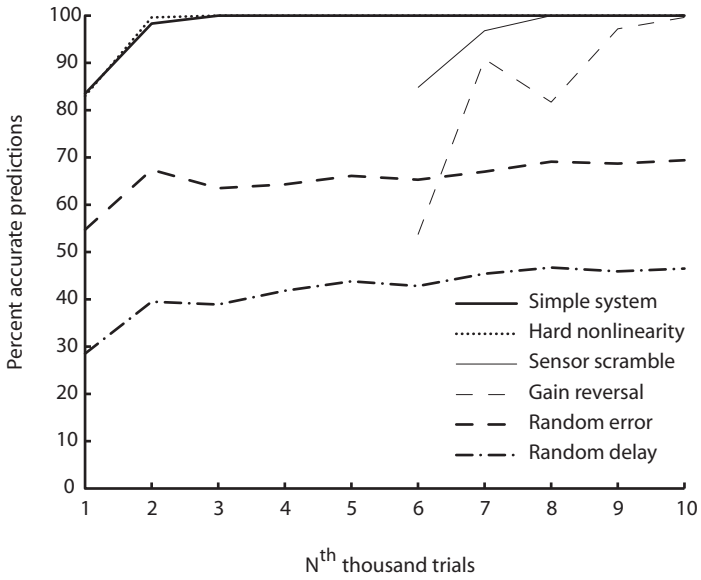

Fig. 2. Simulation performance. Six different conditions were imposed: the simple system (bold solid line), the system with a hard stop (dotted line), the system with scrambled sensory state labels (fine solid line), the system with a command reversal after 5000 trials (fine dashed line), the system with randomness in the movement amplitude (dashed line), and the system with a random time delay (dash-dot line). The scrambled label condition and the gain reversal condition represented perturbations to the simple condition that began during the 6^{th} thousand trials. Prior to that, their performance curves are represented by the simple system curve.

and rarely produced any prediction at all. In the former it was worse than useless; it was misleading. It produced a prediction, but an incorrect one.

The introduction of random noise into the movement amplitude made perfect prediction impossible. The noise amplitude was equal to the resolution of the position measurement, 10°. As a result, knowledge of the current position allowed prediction of the subsequent position with an accuracy of only 50%. With a longer event history, it was possible to increase the accuracy, but only to a certain extent. The learning agent began with a prediction accuracy slightly higher than 50%, and gradually it increased to near 70%, reflecting this.

Random time delays introduced the possibility that zero, one, or two command events might be executed at once. With a 50% probability of delay, at any given time step there was a 25% chance that no command would be executed, a 25% chance that two commands would be executed simultaneously, a 25% chance that the previous command alone would be executed, and only a 25% chance that the current command alone would be executed. As a result, even once the behavior of this simple system is learned, only a 25% success rate can be expected with no knowledge of prior events. However, with a complete knowledge of prior events, it was possible to infer whether

the prior command event had been executed, allowing a theoretical prediction accuracy of 50%. The learning agent began with prediction accuracy slightly higher than 25%, and that accuracy climbed to just over 45% after 10,000 trials.

V. Discussion

The computational requirements of the learning agent were modest. Although the system simulated was simple, the time required for the S-learning algorithm to learn its dynamics fully was less than 30 seconds and data storage requirements were almost negligible ($<$1 MB). While we anticipate that both the learning time and the storage requirements will increase with the number of possible sensory and command events, we do not expect to feel the curse of dimensionality full force. By listing observed patterns in a library, rather than tracking occurrences in a sparse matrix of all conceivable patterns, we expect that storage requirements will grow only modestly with increasing dimensionality. Future research will explore this issue in detail.

Although the pointer robot system simulated here contains only a single degree of freedom for actuation and a single sensor, its implications for modeling human motor control are notable. The S-learning algorithm used did not exploit any knowledge of the robot's structure or dynamics, or even assume continuity or order of the sensor input. Despite these handicaps, the algorithm successfully learned a series of dramatic system disturbances. This demonstration implies that S-learning could also be employed to learn and control a much more complex system, perhaps with the complexity of the human neuromuscular system. The only step necessary to extend S-learning to systems with larger numbers of degrees of freedom is to serialize the various commands and sensor outputs into a single event stream. Otherwise, the algorithm need not be modified.

The large number, nonlinear coupling, and redundancy of degrees of freedom in the human skeletal system do not suit it well to traditional control approaches. In addition, the inherent compliance of muscles inserts an added layer of complexity to the control problem. Attempts to create humanoid robotics typically circumvent these difficulties by creating hardware with more straightforward kinematics and non-backdriveable actuators, but do so at the expense of versatility. For instance, humans' kinematic redundancy allows mechanical impedance to be adapted to address the task at hand, and muscular elasticity allows human movement to respond gracefully to unexpected perturbations and hard non-linearities in the environment. Current high-end humanoid systems have neither of these traits.

Ironically, the high precision and mechanical impedance that make robots more amenable to established control approaches are also high in cost. If a robot's learning and control scheme can tolerate unmodeled joint backlash, nonlinear friction, and structural elasticity, all of which may be environment-, configuration- and load-dependent, then far less expensive hardware can be used to meet the functional requirements of the robot. If S-learning proves its potential to scale to more complex systems, it may lower the barrier to entry for research involving large degree-of-freedom biologically-inspired robots.

Acknowledgment

Sandia is a multiprogram laboratory operated by Sandia Corporation, a Lockheed Martin Company, for the United States Department of Energy under contract DE-AC04-94AL85000.

References

[1] Flash, T. & Hogan, N. (1982). Evidence for an optimization strategy in arm trajectory formation. Society of Neuroscience Abstracts, 8, 282.
[2] Vallbo, A. B., & Wessberg, J. (1993). Organization of motor output in slow finger movements in man. Journal of Physiology, 469, 673-691.
[3] Collewijn, H., Erkelens, C. J., & Steinman, R. M. (1988). Binocular coordination of human horizontal saccadic eye movements. J Physiol, 404, 157-182.
[4] Abend, W., Bizzi, E., & Morasso, P. (1982). Human arm trajectory formation. Brain, 105, 331-348.
[5] Woodworth, R. S. (1899). The accuracy of voluntary movement. Psychology Review Monogr Suppl.
[6] Crossman, E. R. F. W., & Goodeve, P. J. (1983). Feedback control of hand-movements and Fitt's law. Quarterly Journal of Experimental Psychology, 35A, 251-278. (Paper presented at the meeting of the Experimental Psychology Society, Oxford, July 1963)
[7] Doeringer, J. A. (1999). An investigation into the discrete nature of human arm movements. Doctoral dissertation, Massachusetts Institute of Technology.
[8] von Hofsten, C. (1991). Structuring of early reaching movements: A longitudinal study. Journal of Motor Behavior, 23(4), 280-292.
[9] Morasso, P., & Mussa-Ivaldi, F. A. (1982). Trajectory formation and handwriting: A computational model. Biological Cybernetics, 45, 131-142.
[10] Krebs, H. I., Aisen, M. L., Volpe, B. T., & Hogan, N. (1999). Quantization of continuous arm movements in humans with brain injury. Proceedings of the National Academy of Sciences USA, 96, 4645-9. (Neurobiology)
[11] Rohrer, B., Fasoli, S., Krebs, H. I., Volpe, B., Frontera, W. R., Stein, J., & Hogan, N. (2004). Submovements grow larger, fewer, and more blended during stroke recovery. Motor Control, 8(4), 472-483.
[12] Milner, T. E. (1992). A model for the generation of movements requiring endpoint precision. Neuroscience, 49, 365-374.
[13] James, W. (1890). The principles of psychology. New York: Dover Publications.
[14] Stroud, J. M. (1956). The fine structure of psychological time. In H. Quastler (Ed.), Information Theory in Psychology. (p. 174-205). Glencoe IL:Free Press.
[15] Purves, D., Paydarfar, J. A., & Andrews, T. J. (1996). The wagon wheel illusion in movies and reality. Proceedings of the National Academy of Sciences USA, 93, 3693-3697.
[16] Wertheimer, M. (1912). Experimentelle studien über das sehen von bewegung. Z. Psychologie, 61,161-265.
[17] Gho, M., & Varela, F. J. (1988). A quantitative assessment of the dependency of the visual temporal frame upon the cortical rhythm. J Physiol. Paris, 83, 95-101.
[18] VanRullen, R., & Koch, C. (2003). Is perception discrete or continuous? Trends in Cognitive Sciences, 7, 207-213.
[19] Koch, C. (2004). The quest for consciousness. Englewood, Colorado: Roberts & Company Publishers.
[20] Pierce, D., & Kuipers, B. J. (1997). Map learning with uninterpreted sensors and effectors. Artificial Intelligence, 92, 169-227.
[21] Sutton, R. S. & Barto, A.G. (1998). Reinforcement Learning: An Introduction. Cambridge, Massachusetts: The MIT Press.
[22] Watkins, C. J. C. H. (1989). Learning from Delayed Rewards. Doctoral dissertation, Cambridge University, UK.
[23] Grossberg, S. & Paine, R. W. (2000). Attentive learning of sequential handwriting movements: A neural network model. In Sun, R. & Giles C. L. (Eds.), Sequence Learning (p. 349-387). Berlin Heidelberg: Springer-Verlag.
[24] Sun, R. & Giles C. L., Eds. (2000) Sequence Learning. Berlin Heidelberg: Springer-Verlag.